

FS_4412 项目说明手册

华清远见研发中心

2014-09

版权声明

本手册版权归属北京华清远见教育集团研发中心所有，并保留一切权力。非经北京华清远见教育集团研发中心同意，任何单位及个人不得擅自摘录本手册部分或全部，违者我们将追究其法律责任。

目 录

1. 简介	3
1.1 项目简介	3
1.2 FS_4412 开发板外观	3
1.3 FS_2530 模块介绍	5
1.3.1 ZigBee 温湿度传感器	5
1.3.2 ZigBee 风扇模块	6
1.3.4 ZigBee 光敏模块	6
1.3.4 ZigBee 协调器模块	6
1.3.5 协调器模块连接如下所示	7
2 编译源码	8
2.1 编译源码	8
2.1.1 配置交叉工具链	8
2.1.2 编译 Bootloader 源码	8
2.2 编译 linux 内核源码	9
2.2 制作 ext4 文件系统镜像	11
3. 镜像烧写	12
3.1 配置 fastboot 工具	12
3.1.1 准备 fastboot 工具	12
3.1.2 准备链接	12
3.2 启动参数设置	15

1. 简介

1.1 项目简介

智能农场监控系统由 FS_4412 平台、Linux、ZigBee、传感器技术等技术的结合。通过 FS_2530ZigBee 模块采集环境的信息（温度，光照等信息），采集到的信息通过 ZigBee 网络传输到 FS_4412（运行 Linux 系统）。FS_4412 作为智能监控系统的控制系统和服务器，可以实现农场的智能控制和工作人员的远程监控。

图 1 是智能监控系统的概览图。本系统采用 BS 架构，工作人员可以通过 PC、平板、手机等平台登陆客户端，实现农场的远程控制和监控功能。

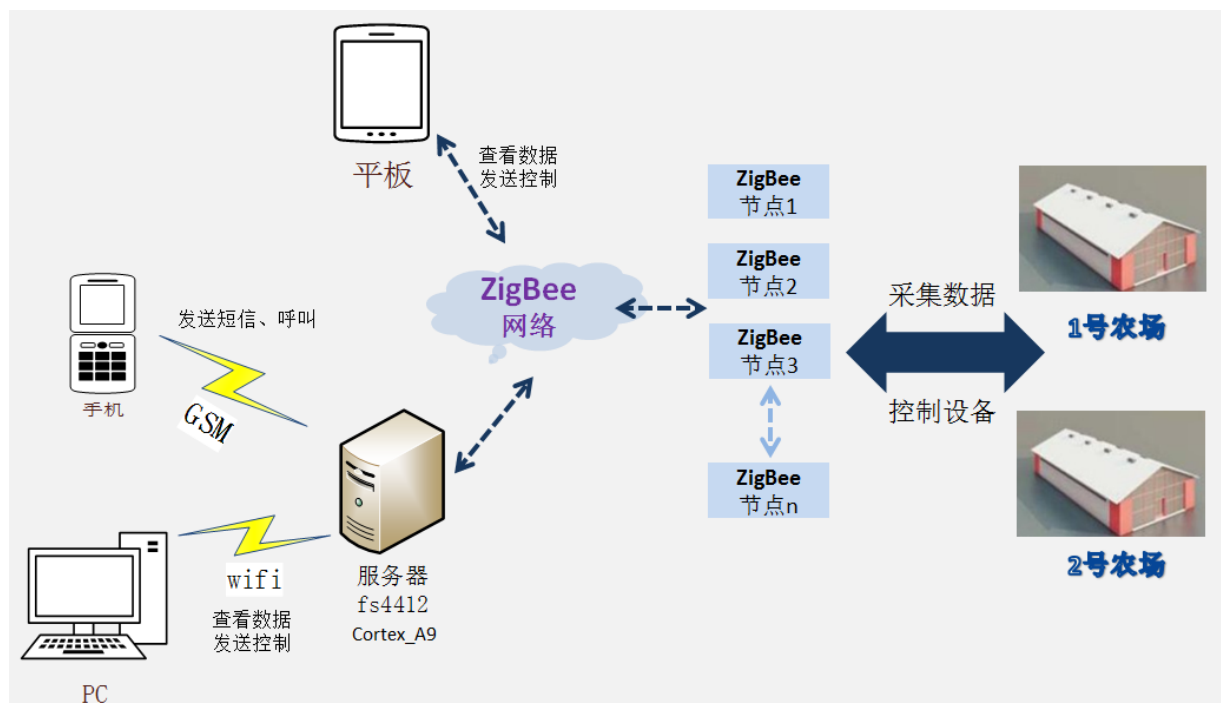


图1 智能监控系统

1.2 FS_4412 开发板外观

图 2 是 FS_4412 开发板的外观。FS_4412 开发平台采用的是高性能四核 Exynos4412 处理器，单核主频达 1.6Ghz，运算能力 16000DMIPS。Exynos4412 采用 32 纳米 HKMG 工艺，在相同性能下，其功耗远胜同类处理器。FS_4412 具有丰富的外设接口，方便开发人员的开

发使用和学习。

Exynos4412 应用领域广泛，FS_4412 平台可以流畅运行 Android4.4 系统。本项目借助 FS_4412 平台开发，运行 Linux 操作系统，作为智能控制系统的服务器和控制中心。

图 2 是 FS_4412 平台图片：

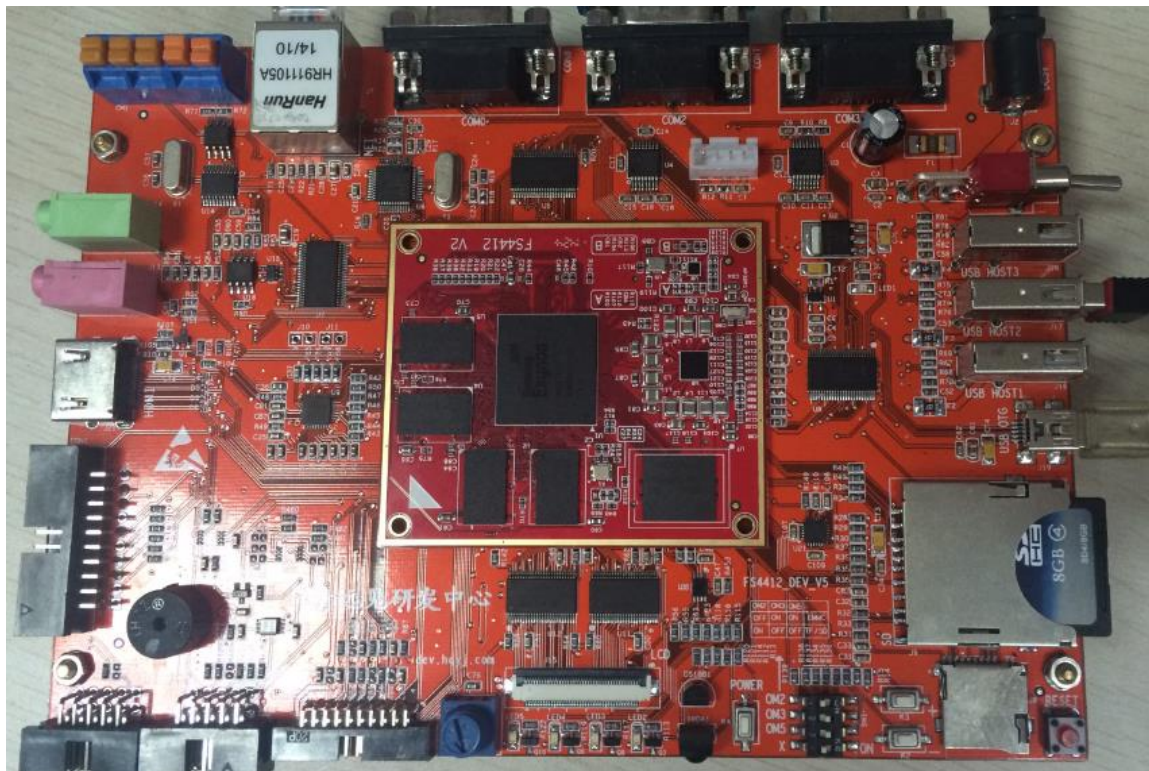


图2 FS_4412 开发板

1.3 FS_2530 模块介绍

智能农场监控系统包含温湿度、光敏、风扇等多个 ZigBee 模块。通过多个 ZigBee 传感器节点模块采集周围环境的相关信息,采集到的信息通过 ZigBee 网络传输到 FS_4412 控制系统。具体的模块介绍如下:

1.3.1 ZigBee 温湿度传感器

FS_2530 温湿度模块,此模块作为智能监控系统的信息采集节点,采集周围环境的温湿度信息,通过 ZigBee 网络将温度、湿度信息传输至控制系统。

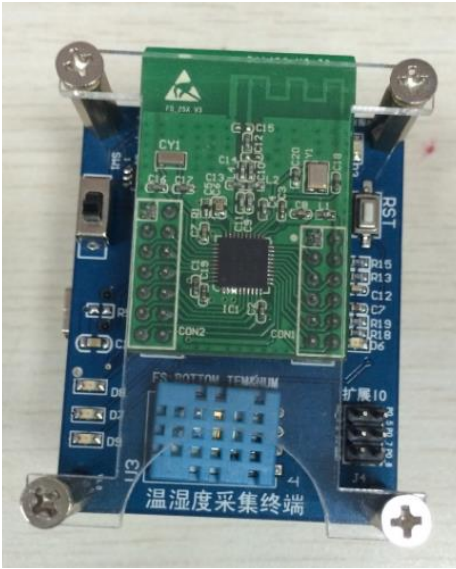


图3 DHT11 温湿度传感器

温湿度传感器		DHT11
相对湿度	分辨率	0.1%RH 16Bit
	重复性	± 1%RH
	精度	25 °C ± 2%RH -40~80 °C ± 5%RH
	互换性	可完全互换
	响应时间	1/e(63%)25 °C 6s 1m/s 空气 6s
	迟滞	< ± 0.3%RH
	长期稳定性	< ± 0.5%RH/yr
温度	分辨率	0.1%RH 16Bit
	重复性	± 0.2 °C
	量程范围	25 °C ± 0.2 °C

1.3.2 ZigBee 风扇模块

FS_2530 风扇模块，此模块作为智能监控系统的执行单元，当控制系统检测出异常情况（例如温度过高）时，FS_4412 控制系统通过协调器节点给执行单元（风扇模块）发送相应的命令，从而采取应急措施。



图4 FS_2530 风扇模块

1.3.4 ZigBee 光敏模块

FS_2530 光敏模块，此模块作为智能监控系统的信息采集节点，采集环境的光照强度信息，通过 ZigBee 网络将光照信息传输至控制系统。



图5 FS_2530 光敏模块

1.3.4 ZigBee 协调器模块

FS_2530 协调器模块，此模块作为智能监控系统的协调器节点，系统中所有节点都与协

调器通信。协调器和 FS_4412 通过 USB 线相连，采集到环境的光照、温度等环境信息都发送给此模块。

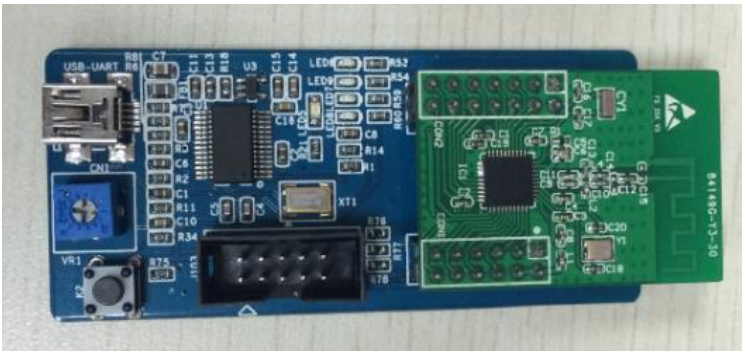


图6 FS_2530 协调器模块

1.3.5 协调器模块连接如下所示

ZigBee 调器模块通过 USB 连接到 FS_4412 的 USB 接口上。如图 7 所示

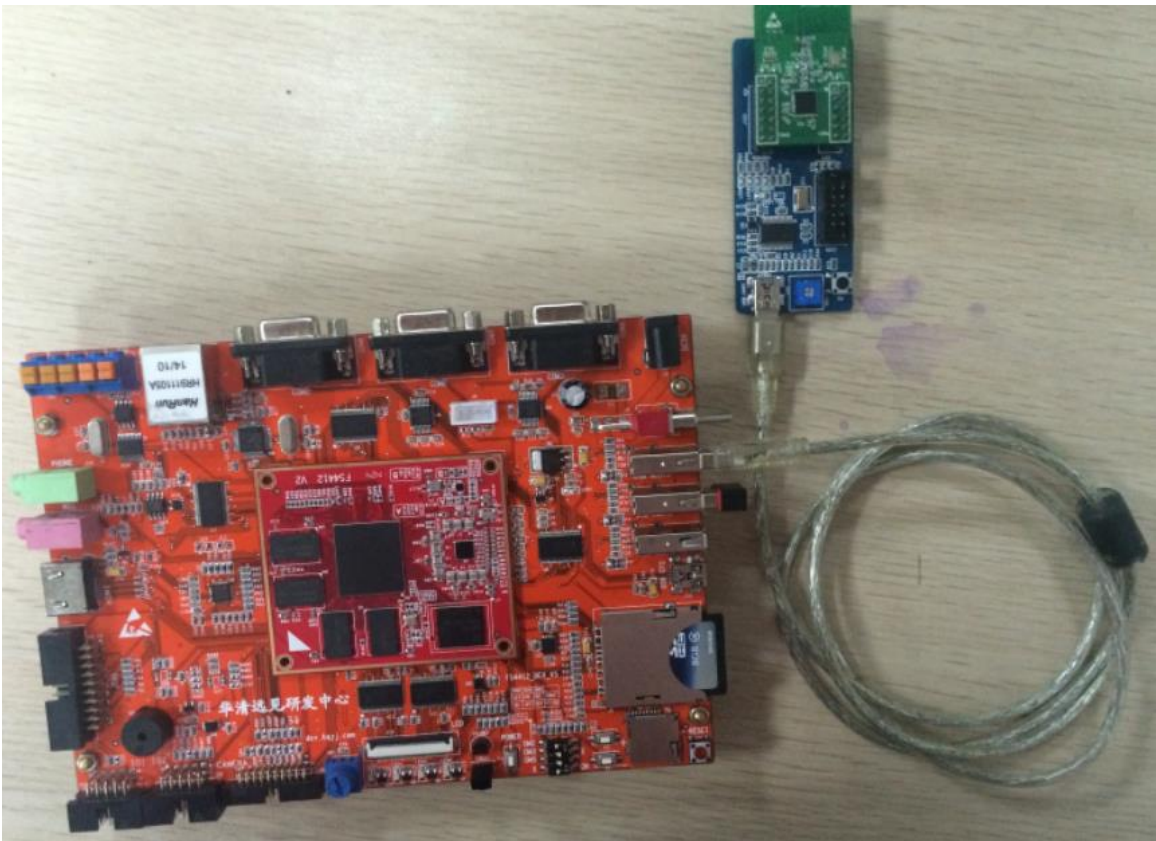


图7 协调器连接

2 编译源码

2.1 编译源码

2.1.1 配置交叉工具链

将“工具”→“Linux”目录中的“toolchain-4.4.6-farsight.tar”文件拷贝到工作目录中。
注意：工作目录可以自行创建，这里以/home/linux/work/4412 为例子
执行以下命令

```
tar xvf toolchain-4.4.6-farsight.tar
```

环境变量的添加

```
sudo vi /etc/bash.bashrc
```

在最后面添加以下命令

```
export PATH=$PATH:/home/linux/work/4412/toolchain-4.4.6/bin/
```

2.1.2 编译 Bootloader 源码

将\Cortex-A9\镜像源码目录中的 uboot-fs4412_v2.tar 拷贝到工作目录中
执行命令：

```
tar xvf uboot-fs4412_v2.tar.xz
```

解压后会出现 uboot-fs4412_v2 目录，进入 /uboot-fs4412_v2/u-boot-2010.03 目录

```
cd /uboot-fs4412_v2/u-boot-2010.03
```

修改 Makefile

```
vi Makefile
```

找到如下图所示代码

```
160 ifeq ($(ARCH),arm)
161 #CROSS_COMPILE = arm-linux-
162 CROSS_COMPILE = /usr/local/toolchain/toolchain-4.4.6/bin/arm-cortex_a8-linux-gnueabi-
163 endif
```

修改为

```
160 ifeq ($(ARCH),arm)
161 #CROSS_COMPILE = arm-linux-
162 CROSS_COMPILE = arm-cortex a8-linux-gnueabi-
163 endif
```

保存退出后执行脚本 build_uboot.sh

```
./ build_uboot.sh
```

编译结果如下图所示

```

linux@ubuntu: ~/work/4412/uboot-fs4412_v2/u-boot-2010.03
File Edit View Terminal Help
cia.a drivers/power/libpower.a drivers/spi/libspi.a drivers rtc/librtc.a drivers/serial/libserial.a drivers/twse
rial/libtws.a drivers/usb/gadget/libusb_gadget.a drivers/usb/host/libusb_host.a drivers/usb/musb/libusb_musb.a d
rivers/usb/phy/libusb_phy.a drivers/video/libvideo.a drivers/watchdog/libwatchdog.a common/libcommon.a libfdt/li
bftd.a api/libapi.a post/libpost.a board/samsung/fs4412/libfs4412.a --end-group /home/linux/work/4412/uboot-fs44
12_v2/u-boot-2010.03/lib arm/eabi_compat.o -L /home/linux/work/4412/toolchain-4.4.6/bin/./lib/gcc/arm-cortex_a8
-linux-gnueabi/4.4.6 -lgcc -Map u-boot.map -o u-boot
arm-cortex_a8-linux-gnueabi-objcopy -O srec u-boot u-boot.srec
arm-cortex_a8-linux-gnueabi-objcopy --gap-fill=0xff -O binary u-boot u-boot.bin
make[1]: Entering directory `/home/linux/work/4412/uboot-fs4412_v2/u-boot-2010.03/sdfuse_q'
gcc -o chksum chksum.c
gcc -o add_sign add_sign.c
gcc -o add_padding add_padding.c
make[1]: Leaving directory `/home/linux/work/4412/uboot-fs4412_v2/u-boot-2010.03/sdfuse_q'
bl2aa file size= 14336B
before padding uboot.bin file size= 266588B
69284 B written

linux@ubuntu:~/work/4412/uboot-fs4412_v2/u-boot-2010.03$ ls

```

下图中的 **u-boot-fs4412.bin** 即是我们所需要的 uboot 文件

```

linux@ubuntu:~/work/4412/uboot-fs4412_v2/u-boot-2010.03$ ls
api                cpu                lib_avr32          lib_nios2          nand_spl          System.map
board              CREDITS            lib_blackfin       lib_ppc            net               tc4_cmm.cmm
build_uboot.sh     disk              libfdt             lib_sh             onenand_ipl       tools
CHANGELOG          doc               lib_generic        lib_sparc          post              u-boot
CHANGELOG-before-U-Boot-1.1.5 drivers            lib_i386           MAINTAINERS        README            u-boot.bin
CodeSign4SecureBoot examples           lib_m68k           MAKEALL            readme.fs4412     u-boot-fs4412.bin
common              fs                lib_microblaze     Makefile           rules.mk          u-boot.lds
config.mk           include           lib_mips            mkconfig           sdfuse            u-boot.map
COPYING             lib_arm           lib_nios            mkuboot.sh         sdfuse_q          u-boot.srec

```

2.2 编译 linux 内核源码

将 \Cortex-A9\ 镜像源码 目录中的 **linux-3.0-fs4412_v3.tar.xz** 拷贝到工作目录中 (/home/linux/work/4412)

执行以下命令解压 linux 源码

```
xz -d linux-3.0-fs4412_v3.tar.xz
```

```

linux@ubuntu:~/work/4412$ ls
linux-3.0-fs4412_v3.tar.xz  project  smart_farm  toolchain-4.4.
linux@ubuntu:~/work/4412$ xz -d linux-3.0-fs4412_v3.tar.xz

```

解压完后, **linux-3.0-fs4412_v3.tar.xz** 解压变成 **linux-3.0-fs4412_v3.tar** 压缩包

```

linux@ubuntu:~/work/4412$ ls
linux-3.0-fs4412_v3.tar  project  smart_farm  toolchain-4.4.6
linux@ubuntu:~/work/4412$

```

再执行以下命令解压 linux-3.0-fs4412_v3.tar 压缩包

```
tar xvf linux-3.0-fs4412_v3.tar
```

解压后如下图所示

```

linux-3.0-fs4412_V3/samples/hidraw/
linux-3.0-fs4412_V3/samples/hidraw/Makefile
linux-3.0-fs4412_V3/samples/hidraw/hid-example.c
linux-3.0-fs4412_V3/samples/tracepoints/
linux-3.0-fs4412_V3/samples/tracepoints/Makefile
linux-3.0-fs4412_V3/samples/tracepoints/tp-samples-trace
linux-3.0-fs4412_V3/samples/tracepoints/tracepoint-probe
linux-3.0-fs4412_V3/samples/tracepoints/tracepoint-samp
linux-3.0-fs4412_V3/samples/tracepoints/tracepoint-probe
linux@ubuntu:~/work/4412$ ls
linux-3.0-fs4412_V3  linux-3.0-fs4412_v3.tar  project  s
linux@ubuntu:~/work/4412$

```

进入 linux 内核目录

```
cd linux-3.0-fs4412_V3
```

修改 Makefile

```
vi Makefile
```

修改 198 行中代码

```
198 CROSS_COMPILE ?= arm-cortex_a8-linux-gnueabi-  
/home/linux/toolchain/toolchain-4.4.6/bin/arm-cortex_a8-linux-gnueabi-
```

```
193 # Note: Some architectures assign CROSS_COMPILE in their arch/*/Makefile  
194 export KBUILD_BUILDHOST := $(SUBARCH)  
195 ARCH ?= arm  
196 #CROSS_COMPILE ?= arm-none-linux-gnueabi-  
197 #CROSS_COMPILE ?= $(CONFIG_CROSS_COMPILE:"%"=%)  
198 CROSS_COMPILE ?= /home/linux/toolchain/toolchain-4.4.6/bin/arm-cortex_a8-linux-gnueabi-  
199  
200 # Architecture as present in compile.h
```

为

```
CROSS_COMPILE ?= arm-cortex_a8-linux-gnueabi-
```

```
194 export KBUILD_BUILDHOST := $(SUBARCH)  
195 ARCH ?= arm  
196 #CROSS_COMPILE ?= arm-none-linux-gnueabi-  
197 #CROSS_COMPILE ?= $(CONFIG_CROSS_COMPILE:"%"=%)  
198 CROSS_COMPILE ?= arm-cortex_a8-linux-gnueabi-  
199
```

修改完后，可以开始编译 linux 源码

执行以下命令：

```
make zImage -j2 //其中“-j2”为编译时使用的 CPU 线程数
```

编译好后如下图所示

```
KSYM .tmp_kallsyms1.S  
AS .tmp_kallsyms1.o  
LD .tmp_vmlinux2  
KSYM .tmp_kallsyms2.S  
AS .tmp_kallsyms2.o  
LD vmlinux  
SYSMAP System.map  
SYSMAP .tmp_System.map  
OBJCOPY arch/arm/boot/Image  
Kernel: arch/arm/boot/Image is ready  
GZIP arch/arm/boot/compressed/piggy.gzip  
SHIPPED arch/arm/boot/compressed/lib1funcs.S  
AS arch/arm/boot/compressed/lib1funcs.o  
AS arch/arm/boot/compressed/piggy.gzip.o  
LD arch/arm/boot/compressed/vmlinux  
OBJCOPY arch/arm/boot/zImage  
Kernel: arch/arm/boot/zImage is ready
```

我们所需要的 linux 内核镜像 **zImage** 在 **arch/arm/boot/** 目录中

2.2 制作 ext4 文件系统镜像

拷贝光盘中\工具\Linux 目录中的 linux_tools.tgz 文件到 Linux 的目录中。

执行解压命令

```
tar xzvf linux_tools.tgz
```

执行命令

```
make_ext4fs -s -l 314572800 -a root -L linux rootfs.img /source/rootfs
```

将文件系统目录（此处是将 nfs 挂载文件系统的文件系统生成镜像）生成文件系统镜像。

注意：其中 `rootfs.img` 为生成的文件系统镜像，`/source/rootfs` 是 nfs 挂载的文件系统目录

```
linux@ubuntu:~/work/4412$ ls
DPO_MT7601U_LinuxSTA_3.0.0.4_20130913  project      toolchain-4.4.6      wifi
linux-3.0-fs4412_V3                  rootfs.img    transplant            wpa_supplicant-0.7.2
openssl-0.9.8e                       smart_farm    u-boot-2013.01-fs4412
linux@ubuntu:~/work/4412$ make_ext4fs -s -l 314572800 -a root -L linux rootfs.img /source/rootfs
Creating filesystem with parameters:
  Size: 314572800
  Block size: 4096
  Blocks per group: 32768
  Inodes per group: 6400
  Inode size: 256
  Journal blocks: 1200
  Label: linux
  Blocks: 76800
  Block groups: 3
  Reserved block group size: 23
Created filesystem with 707/19200 inodes and 17781 76800 blocks
linux@ubuntu:~/work/4412$ ls
DPO_MT7601U_LinuxSTA_3.0.0.4_20130913  project      toolchain-4.4.6      wifi
linux-3.0-fs4412_V3                  rootfs.img    transplant            wpa_supplicant-0.7.2
openssl-0.9.8e                       smart_farm    u-boot-2013.01-fs4412
linux@ubuntu:~/work/4412$
```

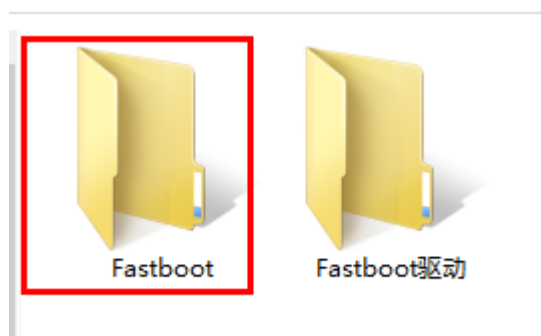
3.镜像烧写

3.1 配置 fastboot 工具

本项目利用 fastboot 工具烧写镜像。在烧写镜像前，我们需要安装相关的 fastboot 驱动。具体步骤如下。

3.1.1 准备 fastboot 工具

fastboot.exe 烧写工具在光盘目录\工具\windows\Fastboot\Fastboot 中。将目录 Fastboot

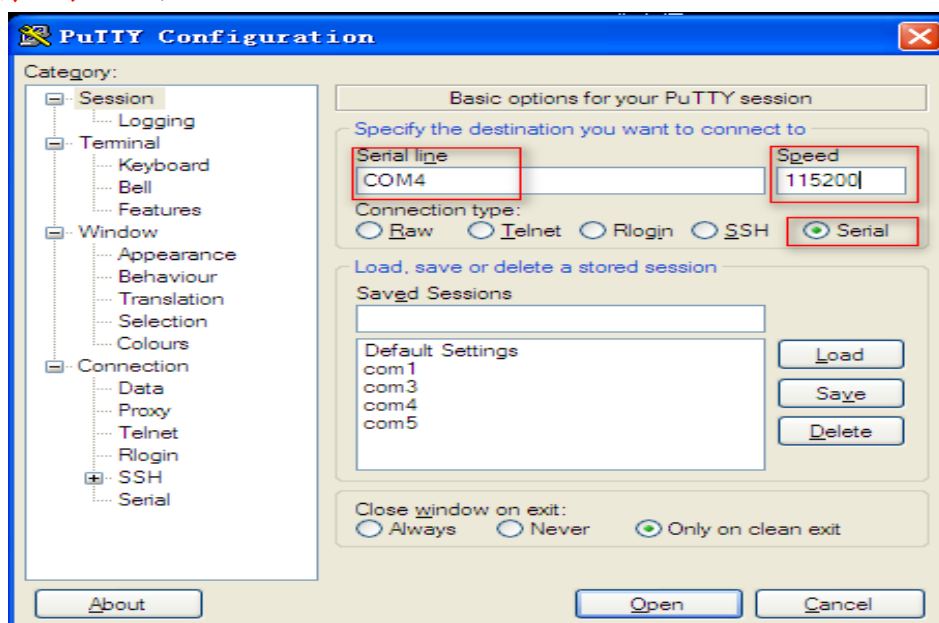


拷贝到 windows 系统的目录下（例如将 Fastboot 目录拷贝到 windows 的 D 盘中）。

3.1.2 准备链接

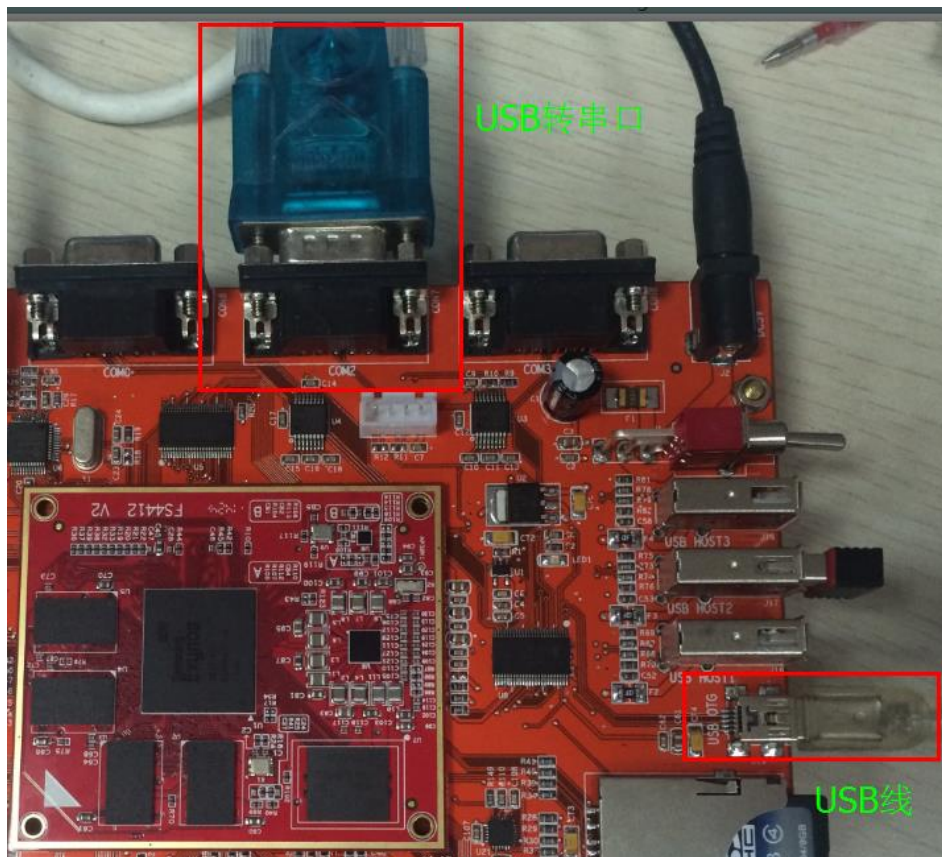
➤ 首先将 USB 转串口线和 USB 线连接到电脑上。

➤ 在光盘\工具\windows 目录中，找到 PUTTY 软件 ，打开 PUTTY 软件并按照下图设置相关参数（注意：端口号 Serial line 根据主机实际的 COM 号配置，本例中端口为 com4）。



配置好后点击 Open 选项。

- 按照下图所示连接好开发板上 USB 转串口线（没有没有装 USB 串口驱动，安装\工具\USB 转串口目录中的 USB 转串口 驱动程序）和 USB 线。



- 打开开发板电源，在倒计时计数到‘0’前按下任意键。

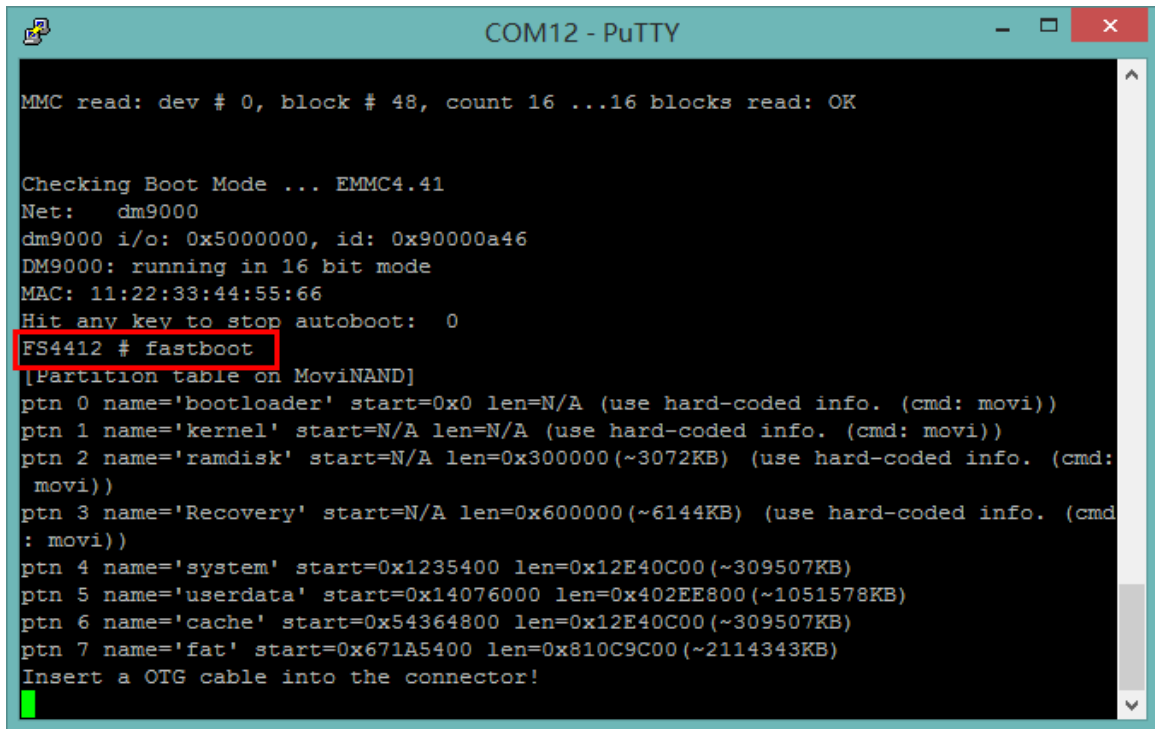
```
SD sclk_mmc is 400K HZ
SD sclk_mmc is 50000K HZ
SD sclk_mmc is 50000K HZ
MMC1: 7536 MB
env_valid = 1
In: serial
Out: serial
Err: serial

MMC read: dev # 0, block # 48, count 16 ...16 blocks read: OK

Checking Boot Mode ... EMMC4.41
Net: dm9000
dm9000 i/o: 0x50000000, id: 0x90000a46
DM9000: running in 16 bit mode
MAC: 11:22:33:44:55:66
Hit any key to stop autoboot: 2
```

倒计时

➤ 然后在 PUTTY 中执行 `fastboot` 命令，如下图所示



进入 fastboot 烧写模式后，如果没有装过 fastboot 驱动，可能会提示安装设备驱动。安装驱动路径为

`\工具\windows\Fastboot\Fastboot 驱动\usb_driver_r03-windows`。具体 fastboot 驱动安装请参考《fastboot 配置参考（辅助）》文档。

将生成的相关镜像拷贝到 windows 下的 Fastboot 目录(之前将 Fastboot 目录拷贝到了 D 盘)。例如，将 bootloader 镜像 `u-boot-fs4412.bin`，内核镜像 `zImage`，文件系统 `rootfs.img` 拷贝到 Fastboot 目录。

名称	修改日期	类型	大小
 adb.exe	2010/6/11 4:13	应用程序	566 KB
 AdbWinApi.dll	2010/6/11 4:13	应用程序扩展	94 KB
 AdbWinUsbApi.dll	2010/6/11 4:13	应用程序扩展	60 KB
 fastboot.exe	2012/3/20 19:03	应用程序	969 KB
 rootfs.img	2014/9/5 20:57	好压 IMG 压缩文件	52,065 KB
 u-boot-fs4412.bin	2014/9/6 22:44	BIN File	515 KB
 zImage	2014/9/6 17:43	文件	3,752 KB

在 windows 的 DOS 终端下，进入 D 盘（之前拷贝 Fastboot 的目录）。

执行以下命令烧写镜像

烧写 bootloader 镜像：

`fastboot.exe flash bootloader u-boot-fs4412.bin`

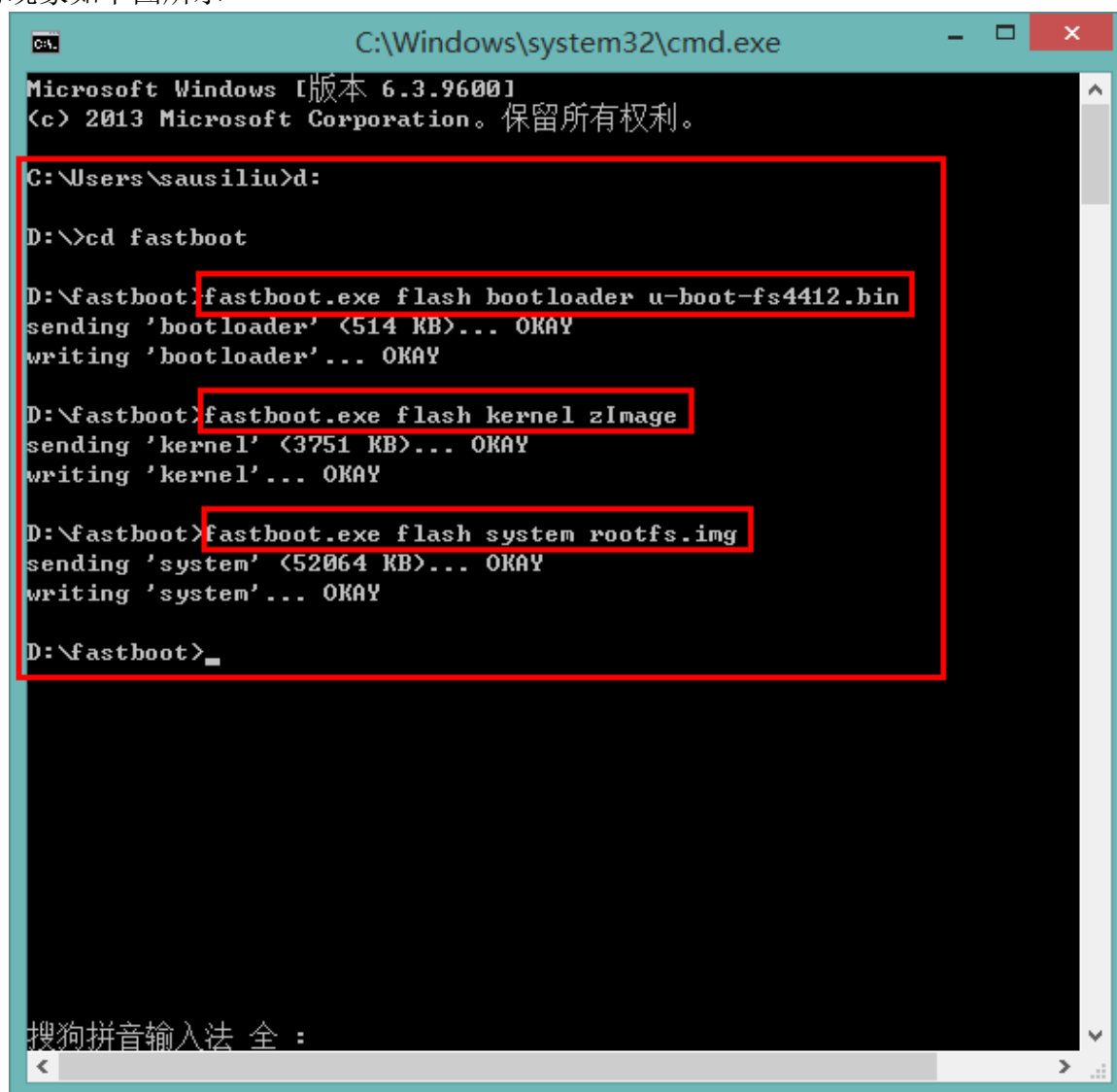
烧写内核镜像：

`fastboot.exe flash kernel zImage`

烧写文件系统镜像

```
fastboot.exe flash system rootfs.img
```

烧写现象如下图所示



```
C:\Windows\system32\cmd.exe
Microsoft Windows [版本 6.3.9600]
(c) 2013 Microsoft Corporation。保留所有权利。

C:\Users\sausiliu>d:
D:\>cd fastboot
D:\fastboot>fastboot.exe flash bootloader u-boot-fs4412.bin
sending 'bootloader' (514 KB)... OKAY
writing 'bootloader'... OKAY
D:\fastboot>fastboot.exe flash kernel zImage
sending 'kernel' (3751 KB)... OKAY
writing 'kernel'... OKAY
D:\fastboot>fastboot.exe flash system rootfs.img
sending 'system' (52064 KB)... OKAY
writing 'system'... OKAY
D:\fastboot>
```

3.2 启动参数设置

镜像烧写完毕后，可以参照以下命令设置启动参数和启动命令

```
setenv bootcmd movi read kernel 40008000\;bootm 40008000
setenv bootargs root=/dev/mmcblk0p2 rw rootfstype=ext4 init=/linuxrc
console=ttySAC0,115200
```